Lab 6 - Classification - DL - Sentiment Analysis

We will focus on **financial texts** and we will use a **BERT** (Bidirectional Enocder Representation from Transformers) based model.

Learning Objectives:

- Run a pretrained BERT model for sentiment classification;
- Apply sentiment analys on financial texts.

Step 1. Setup

```
# Install required libraries
!pip install transformers torch pandas
```

```
seed = 13
```

Step 2. Import the libraries

Task: Check https://huggingface.co/docs/transformers/en/index

```
from transformers import pipeline
import pandas as pd
```

Step 3. Load a pretrained financial sentiment model

Task: Check https://huggingface.co/docs/transformers/en/main_classes/pipelines

```
# sentiment_model = pipeline("sentiment-analysis")
sentiment_model = pipeline("sentiment-analysis", model = "yiyanghkust/finbert-tone")
```

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json:  100%  533/533  [00:00<00:00,  20.0kB/s]
pytorch_model.bin:  100%  439M/439M  [00:10<00:00,  79.6MB/s]
model.safetensors:  100%  439M/439M  [00:13<00:00,  44.2MB/s]
vocab.txt:  226k/?  [00:00<00:00,  2.72MB/s]
Device set to use cuda:0
```

Step 4. Test the approach on a small financial dataset

```
data = {
    "headline": [
        "Apple shares soar after strong quarterly results",
        "Oil prices fall amid concerns over global demand",
        "Tesla announces record deliveries but stock drops",
        "Federal Reserve signals possible interest rate cuts"
    ]
}

df = pd.DataFrame(data)
df
```

	headline
0	Apple shares soar after strong quarterly results
1	Oil prices fall amid concerns over global demand
2	Tesla announces record deliveries but stock drops
3	Federal Reserve signals possible interest rate...

```
#df["headline"].apply(lambda x: sentiment_model(x))
#df["headline"].apply(lambda x: sentiment_model(x))[0] # doar pentru primul rând
#df["headline"].apply(lambda x: sentiment_model(x)[0])[0] # doar pentru primul rând
#df["headline"].apply(lambda x: sentiment_model(x)[0]['label'])[0] # doar pentru primul rând
#df["headline"].apply(lambda x: sentiment_model(x)[0]['label'])
df["headline"].apply(lambda x: sentiment_model(x)[0]['label'].lower())
```

	headline
0	positive
1	negative
2	positive
3	neutral

dtype: object

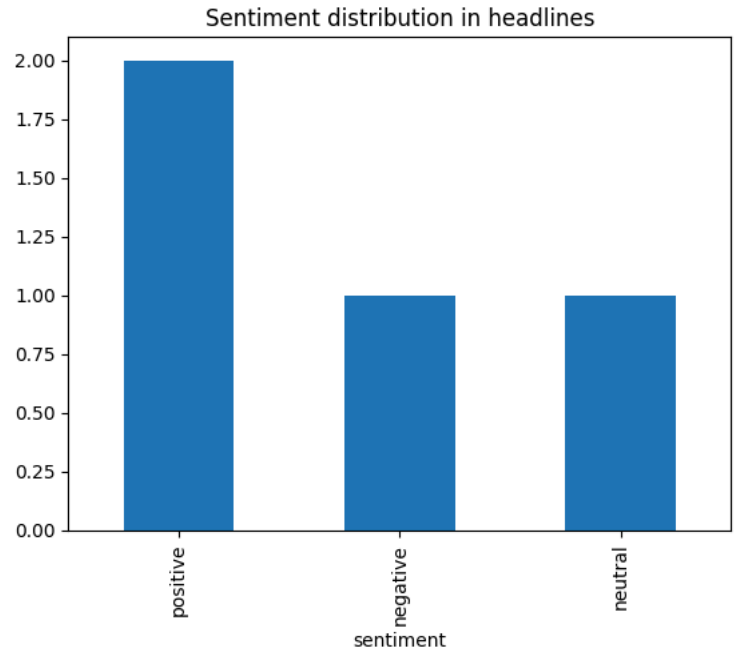
```
# Let's store the results of the sentiment analysis in a column in the df dataset
df["sentiment"] = df["headline"].apply(lambda x: sentiment_model(x)[0]['label'].lower())
df
```

	headline	sentiment
0	Apple shares soar after strong quarterly results	positive
1	Oil prices fall amid concerns over global demand	negative
2	Tesla announces record deliveries but stock drops	positive
3	Federal Reserve signals possible interest rate...	neutral

Step 5. Visualise the results

```
df['sentiment'].value_counts().plot(kind='bar', title='Sentiment distribution in headlines')
```

```
<Axes: title={'center': 'Sentiment distribution in headlines'}, xlabel='sentiment'>
```



Evaluate the transformers' based model on a dataset

```
df = pd.read_csv('/content/Dataset - Classification - Sentences_AllAgree.csv',
                 sep='.@',
                 engine='python')
```

```
df.head()
```

	text	label
0	Russia , although that is where the company is...	neutral
1	For the last quarter of 2010 , Componenta 's n...	positive
2	In the third quarter of 2010 , net sales incre...	positive
3	Operating profit rose to EUR 13.1 mn from EUR ...	positive
4	Operating profit totalled EUR 21.1 mn , up fro...	positive

```
df['label'].value_counts()
```

label	count
neutral	1391
positive	570
negative	303

dtype: int64

The dataframe is unbalanced, with more neutral and positive examples.

```
df_neutral = df[df['label'] == 'neutral']
df_neutral_sampled = df_neutral.sample(303, random_state=seed)

df_postive_sampled = df[df['label'] == 'positive'].sample(303, random_state=seed)
df_negative_sampled = df[df['label'] == 'negative']

df_balanced = pd.concat([df_neutral_sampled, df_postive_sampled, df_negative_sampled])
df_balanced.shape
```

```
(909, 2)
```

```
df_balanced.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 909 entries, 1936 to 2263
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   text    909 non-null    object
 1   label   909 non-null    object
dtypes: object(2)
memory usage: 21.3+ KB
```

```
df_balanced = df_balanced.sample(100, random_state=seed)
df_balanced.shape
```

```
(100, 2)
```

```
# Let's store the results of the sentiment analysis in a column in the df dataset
%time df_balanced["sentiment"] = df_balanced["text"].apply(lambda x: sentiment_model(x)[0]['label']).str.lower()
df_balanced
```

```
You seem to be using the pipelines sequentially on GPU. In order to maximize efficiency please use a dataset
CPU times: user 1.42 s, sys: 841 µs, total: 1.42 s
Wall time: 2.83 s
```

	text	label	sentiment
182	Pretax profit rose to EUR 17.8 mn from EUR 14....	positive	positive
239	Operating profit totaled EUR 825mn , up from E...	positive	positive
130	Ragutis , which is controlled by the Finnish b...	positive	positive
702	The order was valued at over EUR15m	neutral	neutral
460	Finnish software developer Basware Oyj said on...	positive	neutral
...	...	...	...
2090	The result will also be burdened by increased ...	negative	negative
126	For the first nine months of 2010 , the compan...	positive	positive
551	Completion of the transaction is subject to a ...	neutral	neutral
1882	Operating loss of the Pulp & Paper Machinery u...	negative	negative
1592	According to Finnish Scanfil 's founder and ch...	neutral	neutral

100 rows × 3 columns

```
from sklearn.metrics import accuracy_score
acc = accuracy_score(df_balanced['label'], df_balanced['sentiment'])
print(acc)
```

```
0.9
```

```
from sklearn.metrics import classification_report
classification_report(df_balanced['label'], df_balanced['sentiment'])
```

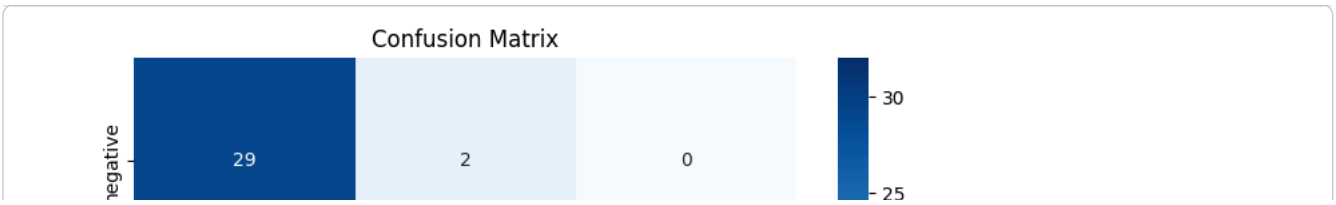
```
'              precision    recall  f1-score   support\n\n    negative       1.00      1.00      1.00         2\n     neutr
al       0.75      1.00      0.86         3\n    positive       1.00      0.80      0.89         5\n\n    accuracy
0.90        10\n   macro avg       0.92      0.93      0.92        10\nweighted avg       0.93      0.90      0.90        1
```

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(df_balanced['label'], df_balanced['sentiment'])
```

```
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix

cm = confusion_matrix(df_balanced['label'], df_balanced['sentiment'])

plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues', xticklabels=['negative', 'neutral', 'positive'], yticklabels=['negative
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.title('Confusion Matrix')
plt.savefig('confusion_matrix.png', dpi=300)
plt.show()
```


Confusion Matrix
29
2
0
30
25